

# Assignment4

22124187 Zhang Linyihan

## 1. Digital Differential Analyzer

```
import matplotlib.pyplot as plt

class DrawLine:

    def DDALine(self, x1, y1, x2, y2):

        dx=x2-x1
        dy=y2-y1

        direction=abs(dx) if abs(dx)>=abs(dy) else abs(dy)
        stepX=float(dx/direction)
        stepY=float(dy/direction)

        pointX=[]
        pointY=[]

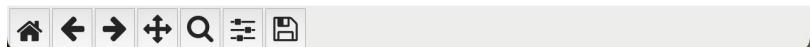
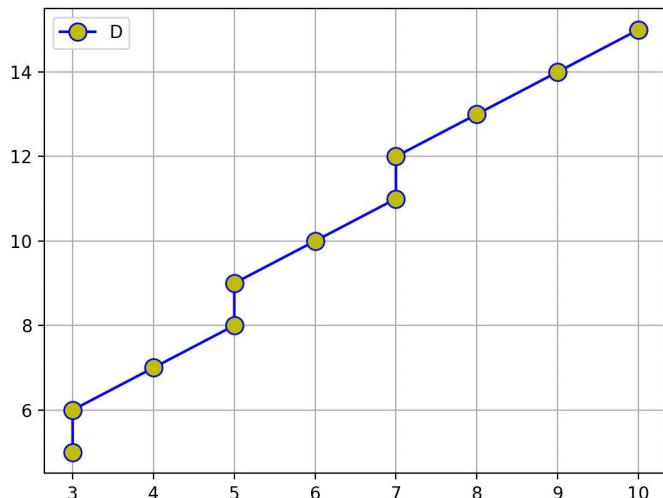
        for i in range(direction+1):
            pointX.append(int(x1))
            pointY.append(int(y1))

            x1+=stepX
            y1+=stepY

        plt.plot(pointX, pointY, color='b', linestyle='-', marker='o', markerfacecolor='y', markersize=10)
        plt.legend('DDA')
        plt.grid(True)
        plt.show()

drawLine=DrawLine()
drawLine.DDALine(3, 5, 10, 15)
```

Figure 1



## 2. Bresenham's line drawing algorithm

```
def BresenhamLine(self, x1, y1, x2, y2):
    pointX = []
    pointY = []
    slope = abs(y2 - y1) > abs(x2 - x1)
    if slope:
        x1, y1 = y1, x1
        x2, y2 = y2, x2

    if x1 > x2:
        x1, x2 = x2, x1
        y1, y2 = y2, y1

    dx = x2 - x1
    dy = abs(y2 - y1)

    error = 0
    step = dy / dx if dx != 0 else 0
    yk = y1

    if y1 < y2:
        y_step = 1
    else:
        y_step = -1

    for xk in range(x1, x2 + 1):
        if slope:
            pointX.append(yk)
            pointY.append(xk)
        else:
            pointX.append(xk)
            pointY.append(yk)

        error += step
        if error >= 0.5:
            yk += y_step
            error -= 1.0

    plt.plot(pointX, pointY, color='b', linestyle='-', marker='o',
             markerfacecolor='y', markersize=5, label='BresenhamLine')
    plt.legend()
    plt.grid(True)
    plt.axis('equal')
    plt.show()

drawLine.BresenhamLine(3, 5, 10, 15)
```

Figure 1

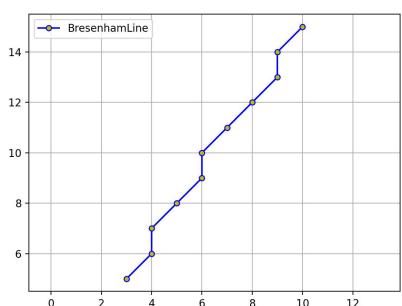


Figure 1